Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 17 of 17 returned.

☐ 1. Document ID: US 6664955 B1

L6: Entry 1 of 17 File: USPT Dec 16, 2003

DOCUMENT-IDENTIFIER: US 6664955 B1
TITLE: Graphics system configured to interpolate pixel values

Detailed Description Text (78):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (95):
FIG. 13 is a flowchart of one embodiment of a method for filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

☐ 2. Document ID: US 6650323 B2

L6: Entry 2 of 17 File: USPT Nov 18, 2003

DOCUMENT-IDENTIFIER: US 6650323 B2
TITLE: Graphics system having a super-sampled sample buffer and having single sample per pixel support

Detailed Description Text (96):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample bu fer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus, the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (115):
FIG. 13 is a flowchart of one embodiment of a method for selecting and filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sam le buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

3. Document ID: US 6624823 B2

L6: Entry 3 of 17 File: USPT Sep 23, 2003

DOCUMENT-IDENTIFIER: US 6624823 B2
TITLE: Graphics system configured to determine triangle orientation by octant identification and slope comparison

Detailed Description Text (101):
FIG. 12A illustrates more detail of one embodiment of a method for reading sample values from super-sampled sample buffer 162. As the figure illustrates, the convolution filter kernel 400 travels across Column I (in the direction of arrow 406) to generate output pixel values, where index I takes any value in the range from one to four. Sample-to-pixel calculation unit 170-I may implement the convolution filter kernel 400. Bin cache 176-I may be used to provide fast access to the memory bins corresponding to Column I. Column I comprises a plurality of bin rows. Each bin row is a horizontal line of spatial bins which stretches from the left column boundary 402 to the right column boundary 404 and spans one bin vertically. In one embodiment, bin cache 176-I has sufficient capacity to store

D.sub.L bin rows of memory bins. The cache line-depth parameter D.sub.L may be chosen to accommodate the support of filter kernel 400. If the support of filter kernel 400 is expected to span no more than D.sub.v bins vertically (i.e. in the Y direction), the cache line-depth parameter D.sub.L may be set equal to D.sub.v or larger.

Detailed Description Text (239):
FIG. 17 is a flowchart of one embodiment of a method for selecting and filtering samples stored in super-sampled sample buffer 162 to generate output pixel values. In step 250, a stream of memory bins are read from the super-sampled sample buffer 162. In step 252, these memory bins may be stored in one or more ƏflYbin aches 176 to allow the sample-to-pixel calculation units 170 easy access to samples (i.e. sample positions and their corresponding ordinate values) during the convolution operation. In step 254, the memory bins are examined to determine which of the memory bins may contain samples that contribute to the output pixel value currently being generated. The support (i.e. footprint) of the filter kernel 400 (see FIG. 12A) intersects a collection of spatial bins. The memory bins corresponding to these samples may contain sample values that contribute to the current output pixel.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

4. Document ID: US 6614445 B1

L6: Entry 4 of 17 File: USPT Sep 2, 2003

DOCUMENT-IDENTIFIER: US 6614445 B1
TITLE: Antialiasing method for computer graphics

Abstract Text (1):
A method and apparatus for improving the quality of images that include aliased pixels is provided. For the method of the present invention, one r more portions of an image are selected for antialiasing. Each selected portion is rendered at a higher than normal resolution into a frame buffer and then read back into as a texture in to a cache or other memory. The texture is then filtered back to its original size. The filtering operation supersamples the texture. The resulting texture is antialiased and anisotropic to a degree that matches the resize. The antialiased texture is then applied to a quadrilateral in the frame buffer.

Brief Summary Text (17):
After rendering each image portion is read back from the frame buffer into a texture memory or texture cache. The image portions are now treated as textures. Each textures is applied to a quadrilateral that is drawn orthogonally to the viewer in the frame buffer. The quadrilaterals are sized to match the normal resolution of the image portions. As a result, the textures (which are created at a higher resolution) must be filtered to fit their quadrilaterals. The texturing hardware performs this process using weighted averages of the pixels included in the textures. The overall result is that the quadrilaterals are supersampled from the original image. This improves antialiasing and ensures that the textures applied to the quadrilaterals are anisotropic to a degree that matches the image resize.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 5. Document ID: US 6577312 B2

L6: Entry 5 of 17 File: USPT Jun 10, 2003

DOCUMENT-IDENTIFIER: US 6577312 B2
TITLE: Graphics system configured to filter samples using a variable support filter

Detailed Description Text (69):
Turning now to FIG. 11B, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. The sample-to-pixel calculation unit assigned to column 414 may implement the convolution filter kernel 400. A bin cache may used to provide quick access to the bins that may potentially contribute samples to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in the bin cache. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of bins and then proceed again. Thus, the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (86):
FIG. 13 is a flowchart of one embodiment of a method for filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 6. Document ID: US 6525723 B1

L6: Entry 6 of 17 File: USPT Feb 25, 2003

DOCUMENT-IDENTIFIER: US 6525723 B1
TITLE: Graphics system which renders samples into a sample buffer and generates pixels in response to stored samples at different rates

Detailed Description Text (84):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to

generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (103):
FIG. 13 is a flowchart of one embodiment of a method for selecting and filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

---

7. Document ID: US 6496187 B1

L6: Entry 7 of 17 File: USPT Dec 17, 2002

DOCUMENT-IDENTIFIER: US 6496187 B1
TITLE: Graphics system configured to perform parallel sample to pixel calculation

Detailed Description Text (113):
Turning now to FIG. 14, more details of one embodiment of a method for reading sample values from a super-sampled sample buffer are shown. As the figure illustrates, the sample-to-pixel filter kernel 400 travels across Column I (in the direction of arrow 406) to generate output pixel values, where index I takes any value in the range from one to four. Sample-to-pixel calculation unit 170-I may implement the sample-to-pixel filter kernel 400. Bin cache 176-I may used to provide fast access to the memory bins corresponding to Column I. For example, bin cache 176-I may have a capacity greater than or equal to 25 memory bins since the support of sample-to-pixel filter kernel 400 covers a 5 by 5 array of spatial bins. As the sample-to-pixel operation proceeds, memory bins are read from the super-sampled sample buffer 162 and stored in bin cache 176-I. In one embodiment, bins that are no longer needed, e.g. bins 410, are overwritten in bin cache 176-I by new bins. As each output pixel is generated, sample-to-pixel filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within Column I in the direction indicated by arrow 406. When kernel 400 reaches the right boundary 404 of the Column I, it may shift down one or more rows of bins, and then, proceed horizontally starting from the left column boundary 402. Thus the sample-to-pixel operation proceeds in a scan line manner generating successive rows of output pixels for display.

Detailed Description Text (136):
FIG. 19 is a flowchart of one embodiment of a method for selecting and filtering samples stored in super-sampled sample buffer 162 to generate output pixel values. In step 250, a stream of memory bins are read from the super-sampled sample buffer 162. In step 252, these memory bins may be stored in one or more of bin caches 176 to allow the sample-to-pixel calculation units 170 easy access to sample values during the sample-to-pixel operation. In step 254, the memory bins are examined to determine which of the memory bins may contain samples that contribute to the output pixel value currently being generated. Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

8. Document ID: US 6496186 B1

L6: Entry 8 of 17 File: USPT Dec 17, 2002

DOCUMENT-IDENTIFIER: US 6496186 B1
**** See image for Certificate of Correction ****
TITLE: Graphics system having a super-sampled sample buffer with generation of output pixels using selective adjustment of filtering for reduced artifacts

Detailed Description Text (84):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the co volution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (103):
FIG. 3 is a flowchart of one embodiment of a method for selecting and filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

9. Document ID: US 6489956 B1

L6: Entry 9 of 17 File: USPT Dec 3, 2002

DOCUMENT-IDENTIFIER: US 6489956 B1
TITLE: Graphics system having a super-sampled sample buffer with generation of output pixels using selective adjustment of filtering for implementation of display effects

Detailed Description Text (84):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (103):
FIG. 13 is a flowchart of one embodiment of a method for selecting and filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

10. Document ID: US 6483504 B1

L6: Entry 10 of 17 File: USPT Nov 19, 2002

DOCUMENT-IDENTIFIER: US 6483504 B1
TITLE: Graphics system having a super sampled-sample buffer with efficient storage of sample position information

Detailed Description Text (99):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (117):
FIG. 13 is a flowchart of one embodiment of a method for selecting and filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution rocess (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

---

11. Document ID: US 6466206 B1

L6: Entry 11 of 17 File: USPT Oct 15, 2002

DOCUMENT-IDENTIFIER: US 6466206 B1
**** See image for Certificate of Correction ****
TITLE: Graphics system with programmable real-time alpha key generation

Detailed Description Text (66):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one. column of output pixels for display.

Detailed Description Text (83):
FIG. 13 is a flowchart of one embodiment of a method for filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

12. Document ID: US 6459428 B1

L6: Entry 12 of 17 File: USPT Oct 1, 2002

DOCUMENT-IDENTIFIER: US 6459428 B1
TITLE: Programmable sample filtering for image rendering

Detailed Description Text (100):
FIG. 11A illustrates more details of one embodiment of a method for reading sample values from super-sampled sample buffer 162. As the figure illustrates, the convolution filter kernel 400 travels across Column I (in the direction of arrow 406) to generate output pixel values, where index I takes any value in the range from one to four. Sample-to-pixel calculation unit 170-I may implement the convolution filter kernel 400. Bin cache 176-I may be used to provide fast access to the memory bins corresponding to Column I. Column I comprises a plurality of bin rows. Each bin row is a horizontal line of spatial bins which stretches from the left column boundary 402 to the right column boundary 404 and spans one bin vertically. In one embodiment, bin cache 176-I has sufficient capacity to store N.sub.L bin rows of memory bins. The cache line-depth parameter N.sub.L may be chosen to accommodate the support of filter kernel 400. If the support of filter kernel 400 is expected to span no more than N.sub.V bins vertically (i.e. in the Y direction), the cache line-depth parameter NL may be set equal to N.sub.L or larger. In the example of FIG. 11A, the filter support covers N.sub.v =5 bins vertically, and the cache line-depth parameter N.sub.L =6.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

13. Document ID: US 6433789 B1

L6: Entry 13 of 17 File: USPT Aug 13, 2002

DOCUMENT-IDENTIFIER: US 6433789 B1
TITLE: Steaming prefetching texture cache for level of detail maps in a 3D-graphics engine

Detailed Description Text (14):
The previous description establishes cache size requirements--mainly prefetching enough texture for two pixel tiles. It may be shown that at most 18 texture blocks are required for each pixel tile pair as long as no super-sampling is employed--see the figures below. The average number required is actually about 10 blocks. Because the prefetch for the next 2 pixel tiles is concurrent with processing of the current pixel tiles, the minimum texture cache size requirement is 32 blocks or 1024 bytes in 16-bit per texel mode. In such cases when 18 blocks are required, the next prefetch only requires 12 additional blocks-as can be seen in FIG. 5B below. Compressed textures may require less; however, they may not be considered exclusively because many applications use uncompressed textures. In tri-linear modes, an additional 8 blocks may be required. However, considering the adaptive tri-linear scheme employed in the invention, the 1K byte cache size is sufficient.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

---

14. Document ID: US 6426755 B1

L6: Entry 14 of 17 File: USPT Jul 30, 2002

DOCUMENT-IDENTIFIER: US 6426755 B1
TITLE: Graphics system using sample tags for blur

Detailed Description Text (69):
Turning now to FIG. 17B, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (86):
FIG. 21 is a flowchart of one embodiment of a method for filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

15. Document ID: US 6424343 B1

L6: Entry 15 of 17 File: USPT Jul 23, 2002

DOCUMENT-IDENTIFIER: US 6424343 B1
TITLE: Graphics system with programmable real-time sample filtering

Detailed Description Text (67):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates, the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin24acche 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (85):
FIG. 13 is a flowchart of one embodiment of a method for filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one r more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

16. Document ID: US 6417861 B1

L6: Entry 16 of 17 File: USPT Jul 9, 2002

DOCUMENT-IDENTIFIER: US 6417861 B1
TITLE: Graphics system with programmable sample positions

Detailed Description Text (67):
Turning now to FIG. 11A, more details of one embodiment of a method for reading the samples from a super-sampled sample buffer are shown. As the figure illustrates,

the convolution filter kernel 400 travels across column 414 (see arrow 406) to generate output pixels. One or more sample-to-pixel calculation units 170 may implement the convolution filter kernel 400. A bin cache 408 may used to provide quick access to the samples that may potentially contribute to the output pixel. As the convolution process proceeds, bins are read from the super-sampled sample buffer and stored in bin cache 408. In one embodiment, bins that are no longer needed 410 are overwritten in the cache by new bins 412. As each pixel is generated, convolution filter kernel 400 shifts. Kernel 400 may be visualized as proceeding in a sequential fashion within the column in the direction indicated by arrow 406. When kernel 400 reaches the end of the column, it may shift down one or more rows of samples and then proceed again. Thus the convolution process proceeds in a scan line manner, generating one column of output pixels for display.

Detailed Description Text (92):
FIG. 13 is a flowchart of one embodiment of a method for filtering samples stored in the super-sampled sample buffer to generate output pixels. First, a stream of bins are read from the super-sampled sample buffer (step 250). These may be stored in one or more caches to allow the sample-to-pixel calculation units 170 easy access during the convolution process (step 252). Next, the bins are examined to determine which may contain samples that contribute to the output pixel currently being generated by the filter process (step 254). Each sample that is in a bin that may contribute to the output pixel is then individually examined to determine if the sample does indeed contribute (steps 256-258). This determination may be based upon the distance from the sample to the center of the output pixel being generated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

17. Document ID: US 5287438 A

L6: Entry 17 of 17 File: USPT Feb 15, 1994

DOCUMENT-IDENTIFIER: US 5287438 A
TITLE: System and method for drawing antialiased polygons

Abstract Text (1):
A system (30) draws antialiased polygons. A CPU (32) is connected to a floating point processor (FPU) (34) by bus (36). The CPU (32) is connected by a 32-bit system bus (38) to a random access memory (RAM) (40), a cache (42) and an interface (44) in graphics subsystem (45). The interface (44) is connected by bus (46) to graphics processor (48). The graphics processor (48) is connected by 120-bit graphics bus (50) to frame buffer (52). The frame buffer (52) is connected to a video digital to analog converter (DAC) (54) by bus (56). The DAC (54) is connected to video display (58) by line (60). The graphics processor (48) use a technique known as super-sampling to combat the effects of aliasing. In aliased mode, the graphics processor (48) use 16 array sites to sample 16 pixels (72). When drawing a polygon or line in antialiased mode, the graphics processor (48) uses the 16 sites to sample at 16 locations (120) within a single pixel (72). The antialiasing is done by determining what proportion of the locations (120) within each pixel (72) are within the polygon and setting a color of each pixel (72) on the basis of the proportion.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
cache same (supersampl$4 or super sampl$4)	17

Display Format: KWIC Change Format

Previous Page Next Page Go to Doc#